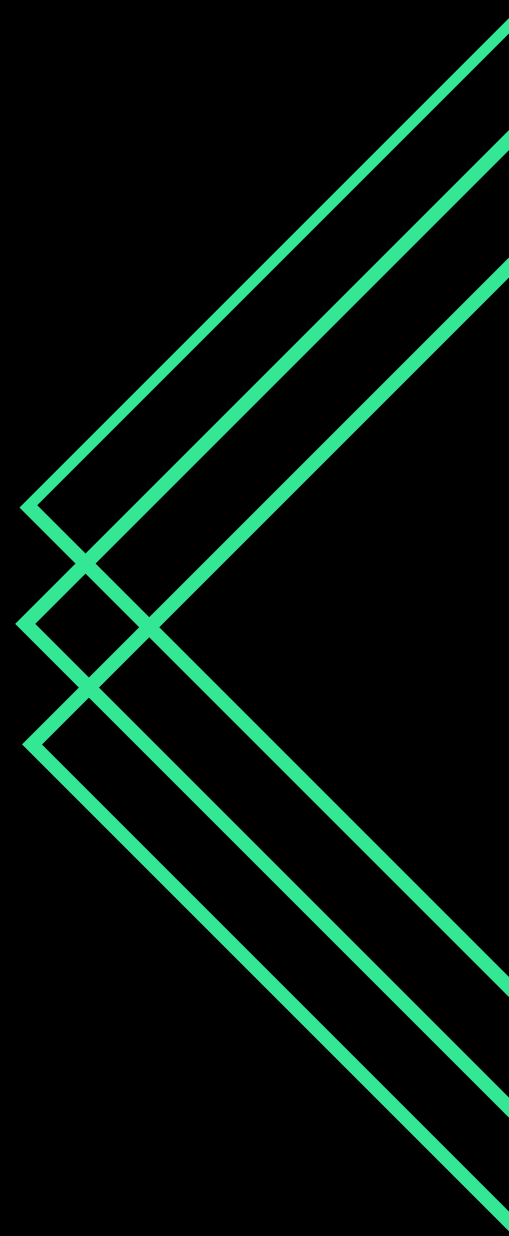





# CREDIT SCORE ENGINE DESIGN

REPORT ON THE DESIGN AND DEVELOPMENT HEURISTICS OF AN  
UNSUPERVISED CREDIT SCORE ENGINE IN PYTHON

Jaspreet Singh Dhani



May 2021

# Problem Statement

- Visualize the data and interpret your thoughts with plots
- Design a credit scoring engine.
- How to improve model and approaches ?
- How to test the model and deployment approaches ?

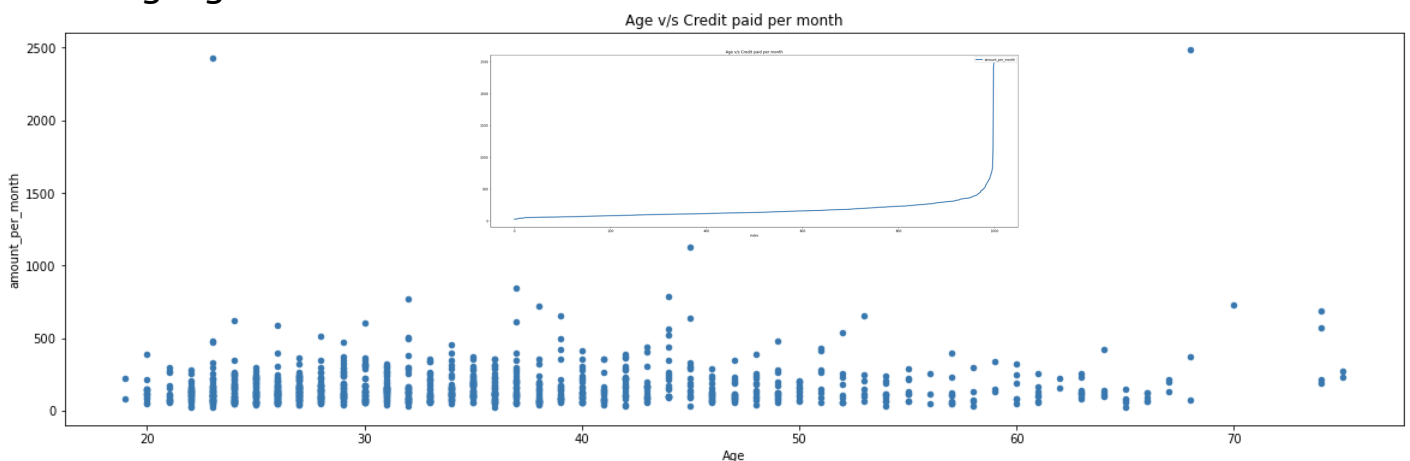


# Data Visualization and Interpretations

# records = **1000**

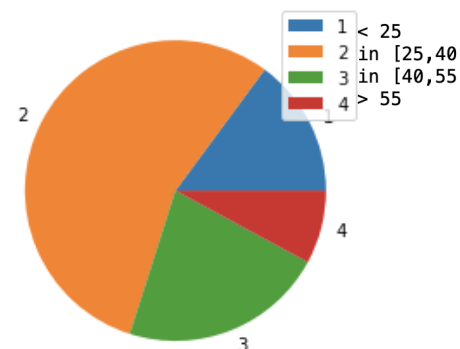
# attributes = **9**

- As a superficial observation, the dataset comprises of no null values in fields except 'Savings account' and 'Checking account'. Thus there is no need to drop certain records due to data sparsity.
- The dataset does not consist of duplicate records, which allows access to a more diverse dataset. However, given there are 9 attributes, with majority of them being multi-labeled, the dataset can be of a greater size to analyze results for more combinations of attribute values.
- The dataset is noise-free with no requirement for pre-processing and data wrangling.



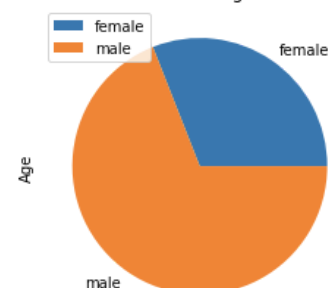
- The scatter plot above depicts the distribution of ratio of credit amount and duration with age. As can be observed, few outliers exist in the given dataset. Also it can be observed that majority of records follow a consistency with the credit ratio ( as can be observed from the line plot above ), with a large difference in ratio value between such records and the outliers. Thus the credit scoring engine must be implemented in a way that it normalizes such ambiguities to the fullest.

Age group distribution in the dataset



- The pie chart ( right and below the scatter plot ), depicts that majority of records are of people in the age group of 25-40 years. This, along with the observation that the data is Male dominated ( pie chart on the bottom right ), implies that model must account for their priorities from a financial perspective for a good result set.

Gender distribution throughout the dataset



# Credit Score Engine

*The Credit Score Engine is based on the concepts of **feature representation, clustering** and foundations of **mathematics and finance**.*

## Data Preparation

### STEP 1: ORDINAL ENCODING OF CATEGORICAL ATTRIBUTES

The ordinal encoding was implemented keeping in mind the relative priorities of the categorical values. For instance, a person with 'rich' savings account will have a higher priority number than a person with 'little' or no savings account at all.

```
1 housing = {'rent':1, 'free':2, 'own':3} #these values are identified using data.Housing.unique()
2
3 sex = {'female':1, 'male':2} #these values are identified using data.Sex.unique()
4
5 saving = {'None':-1, 'little':1, 'moderate':2, 'rich':3, 'quite rich':4} #these values are identified using data[
6
7 checking = {'None':-1, 'little':1, 'moderate':2, 'rich':3} #these values are identified using data['Checking ac
```

Since there was no specific description of job labels, thus based on statistics of the average credit paid per month by people of each job category, the priorities have been assigned.

### STEP 2: CATEGORIZING AGE INTO AGE GROUPS

Four age groups are created and each age value is then an age group accordingly. For eg: An age value of 30 lies in the age group 25-40.

The four age groups specified for the model are:

1) <25    2) 25-40    3) 40-55    4) 55+

These labels have also been ordinal encoded for ease of computation in the later stages of inferential analysis.

```
1 # computing the maximum and minimum age values in the given dataset.
2
3 age_min = df['Age'].min()
4 age_max = df['Age'].max()
5
6 #creating age groups of the below given range
7 breakpoints = [age_min-1, 25, 40, 55, age_max+1]
8
9 # age group labels
10 age_groups = ['1', '2', '3', '4']
11
12 #creating a new attribute with value as the age group label of the given age
13 df['age_group'] = pd.cut(df['Age'], bins=breakpoints, labels=age_groups, right=False)
```

## **STEP 3: DEFINING ADDITIONAL DATA ATTRIBUTES**

A few additional features have been designed to better encapsulate the provided information in the dataset. These features are as follows:

### **1) Average credit amount per month = Credit Amount / Duration**

- The higher this ratio, the more is the contribution of this attribute towards a better credit score.

### **2) Geometric Plane**

- Using the 4 possible binary combinations of Savings account and Checking account values ( i.e. either the account is NA or has a label in [ 'little', 'moderate' ... ] etc, the entire dataset is partitioned into 4 geometric planes. The underlying notion behind this approach is explained in the following section.


## **Approach**

**The model aims to compute the credit score by first computing a penalty score and then normalizing it in the range of 300-900 ( credit score range ).**

### **Computing Penalty Score**

- The penalty score is computed by considering each record as a point in the 3D coordinate space. This space is assumed to comprise of 4 planes stack one over the other.
- Each plane defines a coordinate as ( *age\_group* , *average\_credit\_per\_month* )
- The planes are stacked in decreasing order of priorities, i.e. the top-most plane has the highest priority.
- Each record appears in exactly one of these 4 geometric planes, depending on its values in Savings account and Checking account.
- The overall penalty score is computed using the sum of penalty score for each attribute value of the record. ( *Formula and diagram on the next page* )

### **Penalty Score Normalization and Credit Score Generation**

- Penalty Score holds an inverse relationship with Credit Score.
  - To avoid the negative impact of the presence of outliers in the data, the penalty values were divided into quintiles. Each penalty quartile corresponds to the credit score quintiles in the range 300-900. ( i.e. 300-420 , 420-540.... )
  - For each quintile, the credit score was computed using the formula
- 

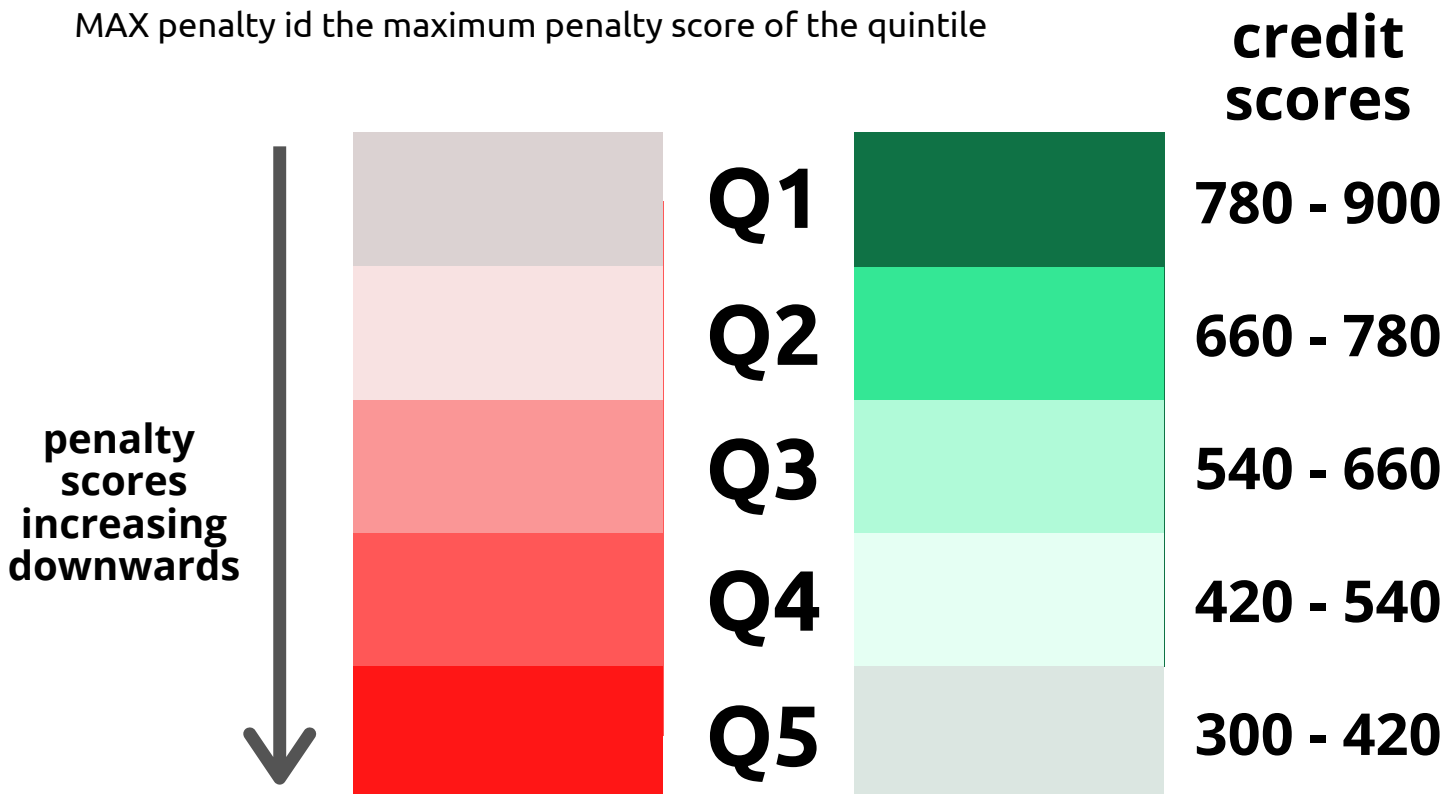
$$\text{credit} - ((\text{penalty score of the record}) / \text{MAX}) * (\text{MAX} - \text{MIN})$$

credit
penalty
credit
credit

MAX credit is the maximum credit score of the quintile

MIN credit is the minimum credit score of the quintile

MAX penalty id the maximum penalty score of the quintile



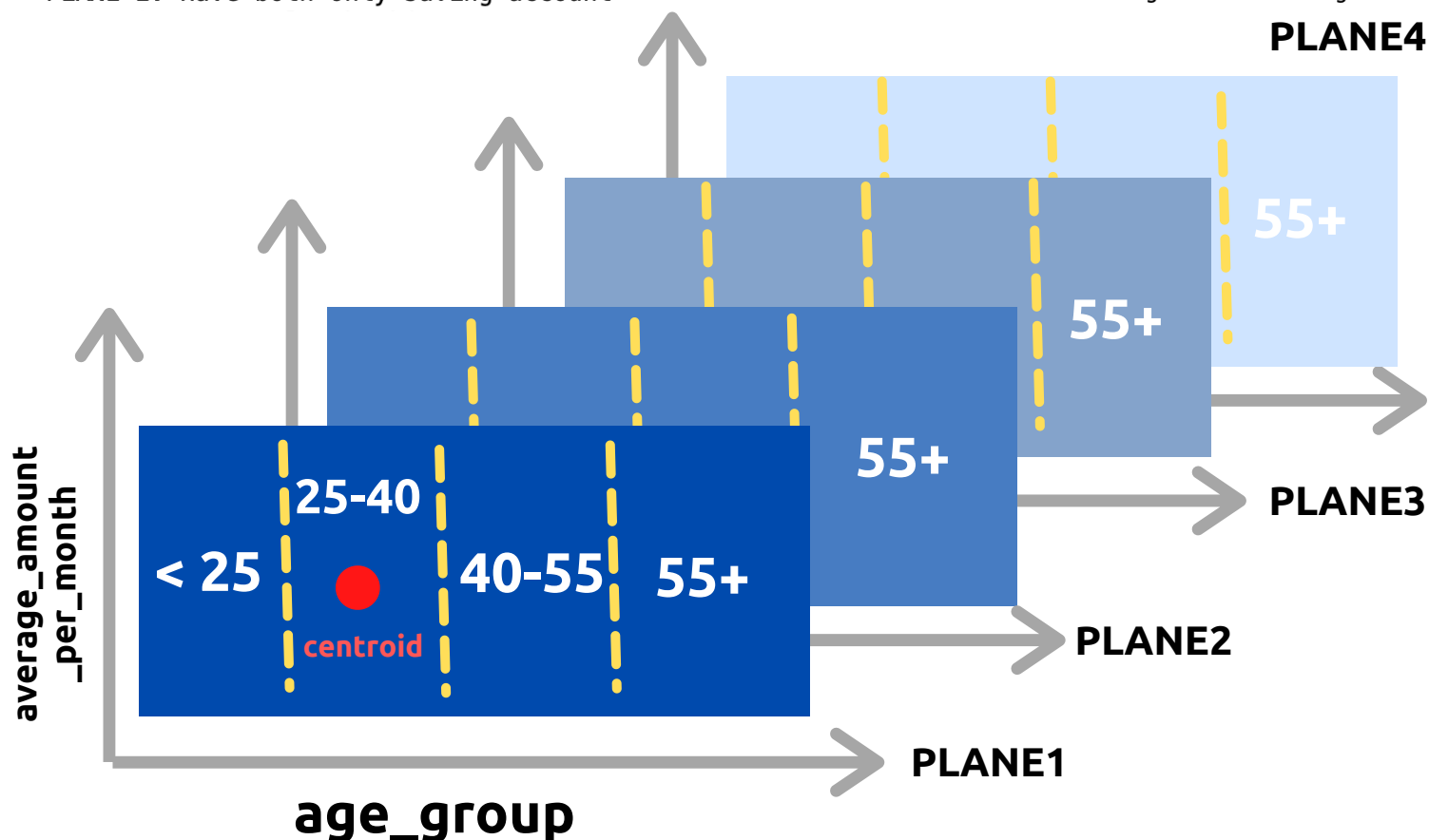
### Visualizing and formulating penalty score computation

PLANE 1: Have both savings and checking accounts

PLANE 2: Have both only saving account

PLANE 3: Have only checking account

PLANE 4: Have neither savings nor checking account



**penalty score for a record =**

$$\begin{aligned} & [ \text{(Euclidean distance from centroid} \\ & \quad * \\ & \quad \text{penalty due to age group)} \\ & \quad + \\ & \text{penalty ( or benefit ) due to average credit amount per} \\ & \quad \text{month} \\ & \quad + \\ & \quad \text{penalty due to housing type} \\ & \quad + \\ & \quad \text{penalty due to saving account status} \\ & \quad + \\ & \text{penalty due to checking account status} \\ & \quad + \\ & \quad \text{penalty due to gender} \\ & \quad + \\ & \quad \text{penalty due to job } ] \\ & \quad * \\ & \text{penalty weight} \end{aligned}$$

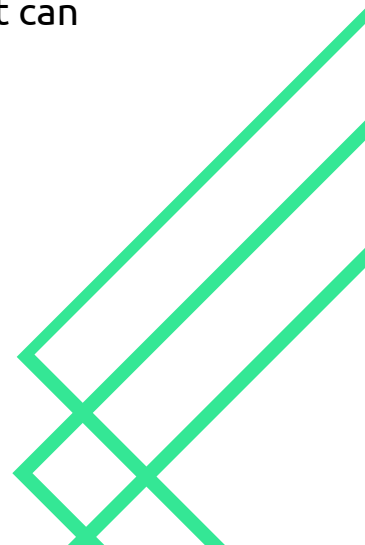
\* penalty score to credit score conversion  
formula on previous page



# How to improve the model and approaches ?

The current implementation of credit score engine can benefit from improving upon certain currently incorporated approaches in the following manners:

- 1) **Better tuning of penalty parameters:** The penalty weights currently deployed in the model are simply the differences between the ordinal encoded labels of the attributes. A better feature correlation can help in tuning such penalty parameters in a much more effective manner.
- 2) **Generalizing model for 'Purpose' attribute:** The purpose of credit does not play any role in this model. A better utilization of this model to benefit/ penalize the 'Purpose' can benefit in improving the performance of the model.
- 3) **Supervised approach:** A database of pre-recorded credit scores can help in designing a supervised approach of the credit score engine. The model gets access to data and can leverage this information to tune its parameters for improved performance. A regression based approach can be designed in such a scenario.
- 4) **Greater dataset size:** Since the current dataset comprises of only 1000 records and with an appreciable number of attributes, the model has the tendency to ingest much more data and produce results for a much more set of inputs. A thorough analysis of a greater dataset can help in better generalization of the credit score engine.
- 5) **Access to payment history:** By analyzing the payment history patterns for each user, the model can take this pattern into consideration in order to better compute the credit score.





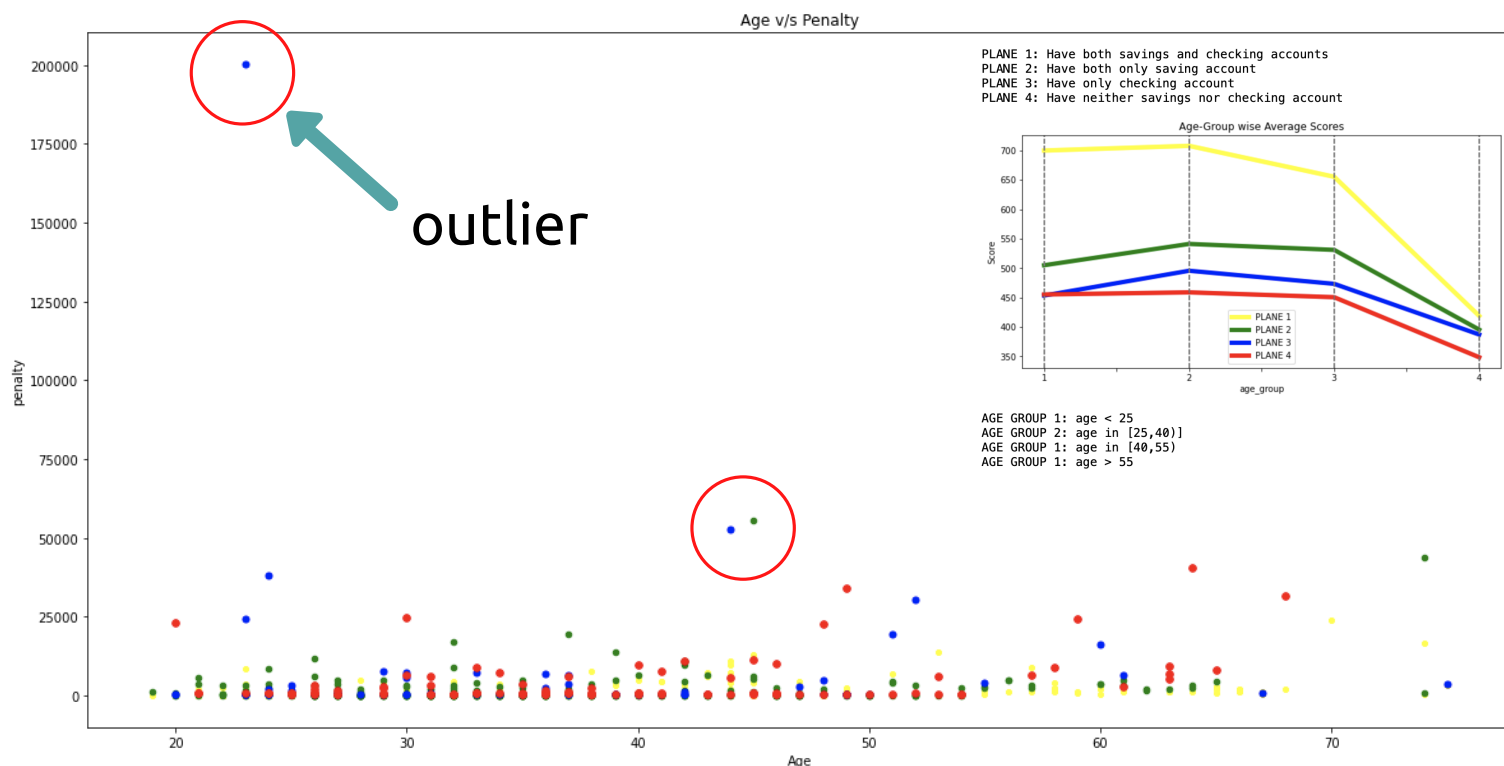
# How to test the model and deployment approaches ?

In order to test the performance of the model, the following methodologies can be adopted:

1) **Checking the obvious trends:** Attributes such as savings account and checking account status play a crucial role in depicting credit scores. If a person doesn't have either of them, then the credit score is highly likely to be lesser than a person with both the accounts.

40	female	2	own	None	None	894	10	education	490
35	male	1	own	rich	moderate	1941	18	business	840

Other obvious trends include to cross-check whether the outliers are not degrading the model's performance.



As can be observed from the line graph above, the younger age groups have a higher average credit score than the elders. The drastic dip in the scores occurs for elder age group due to accumulated effect of rest of the attributes.

2) **Deploying the model on labeled dataset:** Having access to a labeled dataset of appreciable size can help in computing the accuracy and other performance metrics for this implementation of the credit score engine.

3) **Testing the model's performance on extreme ends:** Given a set of records with extreme attribute values, ( for eg: a record for a rented owner type female of age 70 with no saving and checking account and a credit amount of 50000 in a duration of 2 months. ), the model must return the obvious results in order to pass the basic necessities as a working implementation ready for further stringent analysis.

4) **Comparing results with other credit score engine implementations:** The same dataset can be fed to multiple implementations of the credit score model and their respective results can be analyzed for comparison. This way, the models' accuracy can be determined from an unsupervised perspective. From a supervised point of view, it's only a matter compare the accuracy of each model over the same labeled dataset.

5) **Financial expertise:** The supervision of a financial expert can help determine the patterns in the model's inconsistencies and ambiguities. The expertise can also help in suggesting improvements to the model's parameters as per the priorities of data attributes.

-----

